Network Working Group   J. Lennox
Internet-Draft   Vidyo
Intended status: Informational   K. Gross
Expires: July 20, 2015   AVA
                                                   S. Nandakumar
                                                     G. Salgueiro
                                                    Cisco Systems
                                                        B. Burman
                                                         Ericsson
                                                 January 16, 2015

A Taxonomy of Grouping Semantics and Mechanisms for Real-Time Transport
                        Protocol (RTP) Sources
               draft-ietf-avtext-rtp-grouping-taxonomy-04

Abstract

   The terminology about, and associations among, Real-Time Transport
   Protocol (RTP) sources can be complex and somewhat opaque. This
   document describes a number of existing and proposed relationships
   among RTP sources, and attempts to define common terminology for
   discussing protocol entities and their relationships.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on July 20, 2015.

Table of Contents

1. Introduction

The existing taxonomy of sources in RTP is often regarded as
confusing and inconsistent. Consequently, a deep understanding of
how the different terms relate to each other becomes a real
challenge. Frequently cited examples of this confusion are (1) how
different protocols that make use of RTP use the same terms to
signify different things and (2) how the complexities addressed at
one layer are often glossed over or ignored at another.

This document attempts to provide some clarity by reviewing the
semantics of various aspects of sources in RTP. As an organizing
mechanism, it approaches this by describing various ways that RTP
sources can be grouped and associated together.

All non-specific references to ControLling mUltiple streams for
tElepresence (CLUE) in this document map to [I-D.ietf-clue-framework]
and all references to Web Real-Time Communications (WebRTC) map to
[I-D.ietf-rtcweb-overview].

2. Concepts

This section defines concepts that serve to identify and name various
transformations and streams in a given RTP usage. For each concept
an attempt is made to list any alternate definitions and usages that
co-exist today along with various characteristics that further
describes the concept. These concepts are divided into two
categories, one related to the chain of streams and transformations
that media can be subject to, the other for entities involved in the
communication.

2.1. Media Chain

In the context of this memo, Media is a sequence of synthetic or
Physical Stimulus (Section 2.1.1) (sound waves, photons, key-
strokes), represented in digital form. Synthesized Media is
typically generated directly in the digital domain.

This section contains the concepts that can be involved in taking
Media at a sender side and transporting it to a receiver, which may
recover a sequence of physical stimulus. This chain of concepts is
of two main types, streams and transformations. Streams are time-
based sequences of samples of the physical stimulus in various
representations, while transformations changes the representation of
the streams in some way.

The below examples are basic ones and it is important to keep in mind
that this conceptual model enables more complex usages. Some will be
further discussed in later sections of this document. In general the
following applies to this model:

o A transformation may have zero or more inputs and one or more
   outputs.

o A stream is of some type, such as audio, video, real-time text,
   etc.

o A stream has one source transformation and one or more sink
   transformations (with the exception of Physical Stimulus
   (Section 2.1.1) that may lack source or sink transformation).

o Streams can be forwarded from a transformation output to any
   number of inputs on other transformations that support that type.

o If the output of a transformation is sent to multiple
   transformations, those streams will be identical; it takes a
   transformation to make them different.

o There are no formal limitations on how streams are connected to
   transformations, this may include loops if required by a
   particular transformation.

It is also important to remember that this is a conceptual model.
Thus real-world implementations may look different and have different
structure.

To provide a basic understanding of the relationships in the chain we
below first introduce the concepts for the sender side (Figure 1).
This covers physical stimulus until media packets are emitted onto
the network.

```
               Physical Stimulus
                      |
                      V
          +--------------------+
          |    Media Capture   |
          +--------------------+
                      |
                 Raw Stream
                      V
          +--------------------+
          |    Media Source    |<- Synchronization Timing
          +--------------------+
                      |
                Source Stream
                      V
          +--------------------+
          |   Media Encoder    |
          +--------------------+
                      |
             Encoded Stream     +------------+
                      V         |            V
          +--------------------+ | +----------------------+
          |  Media Packetizer  | | | RTP-based Redundancy |
          +--------------------+ | +----------------------+
                      |          |            |
                      +------------+  Redundancy RTP Stream
             Source RTP Stream                |
                      V                        V
          +--------------------+    +--------------------+
          |  Media Transport   |    |  Media Transport   |
          +--------------------+    +--------------------+
```
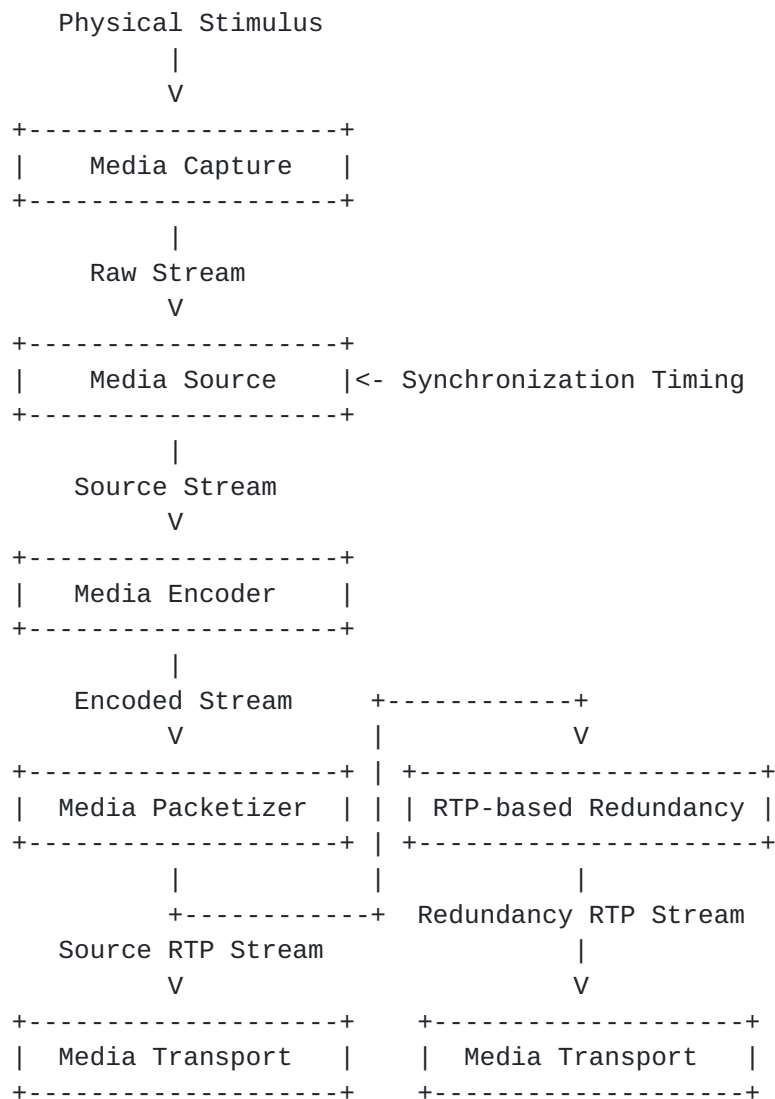
                Figure 1: Sender Side Concepts in the Media Chain

   In Figure 1 we have included a branched chain to cover the concepts
   for using redundancy to improve the reliability of the transport.
   The Media Transport concept is an aggregate that is decomposed below
   in Section 2.1.13.

   Below we review a receiver media chain (Figure 2) matching the sender
   side, to look at the inverse transformations and their attempts to
   recover identical streams as in the sender chain, subject to what may
   be lossy compression and imperfect Media Transport. Note that the
   streams out of a reverse transformation, like the Source Stream out
   the Media Decoder are in many cases not the same as the corresponding
   ones on the sender side, thus they are prefixed with a "Received" to
   denote a potentially modified version. The reason for not being the
   same lies in the transformations that can be of irreversible type.

For example, lossy source coding in the Media Encoder prevents the
Source Stream out of the Media Decoder to be the same as the one fed
into the Media Encoder. Other reasons include packet loss or late
loss in the Media Transport transformation that even RTP-based
Repair, if used, fails to repair. It should be noted that some
transformations are not always present, like RTP-based Repair that
cannot operate without Redundancy RTP Streams.

```
        +--------------------+   +--------------------+
        |  Media Transport   |   |  Media Transport   |
        +--------------------+   +--------------------+
                 |                        |
          Received RTP Stream  Received Redundancy RTP Stream
                 |                        |
                 |     +-------------------+
                 V     V
        +--------------------+
        |  RTP-based Repair  |
        +--------------------+
                 |
          Repaired RTP Stream
                 V
        +--------------------+
        | Media Depacketizer |
        +--------------------+
                 |
        Received Encoded Stream
                 V
        +--------------------+
        |   Media Decoder    |
        +--------------------+
                 |
        Received Source Stream
                 V
        +--------------------+
        |     Media Sink     |--> Synchronization Information
        +--------------------+
                 |
          Received Raw Stream
                 V
        +--------------------+
        |   Media Renderer   |
        +--------------------+
                 |
                 V
           Physical Stimulus
```

          Figure 2: Receiver Side Concepts of the Media Chain

2.1.1. **Physical Stimulus**

The physical stimulus is a physical event that can be sampled and
converted to digital form by an appropriate sensor or transducer.
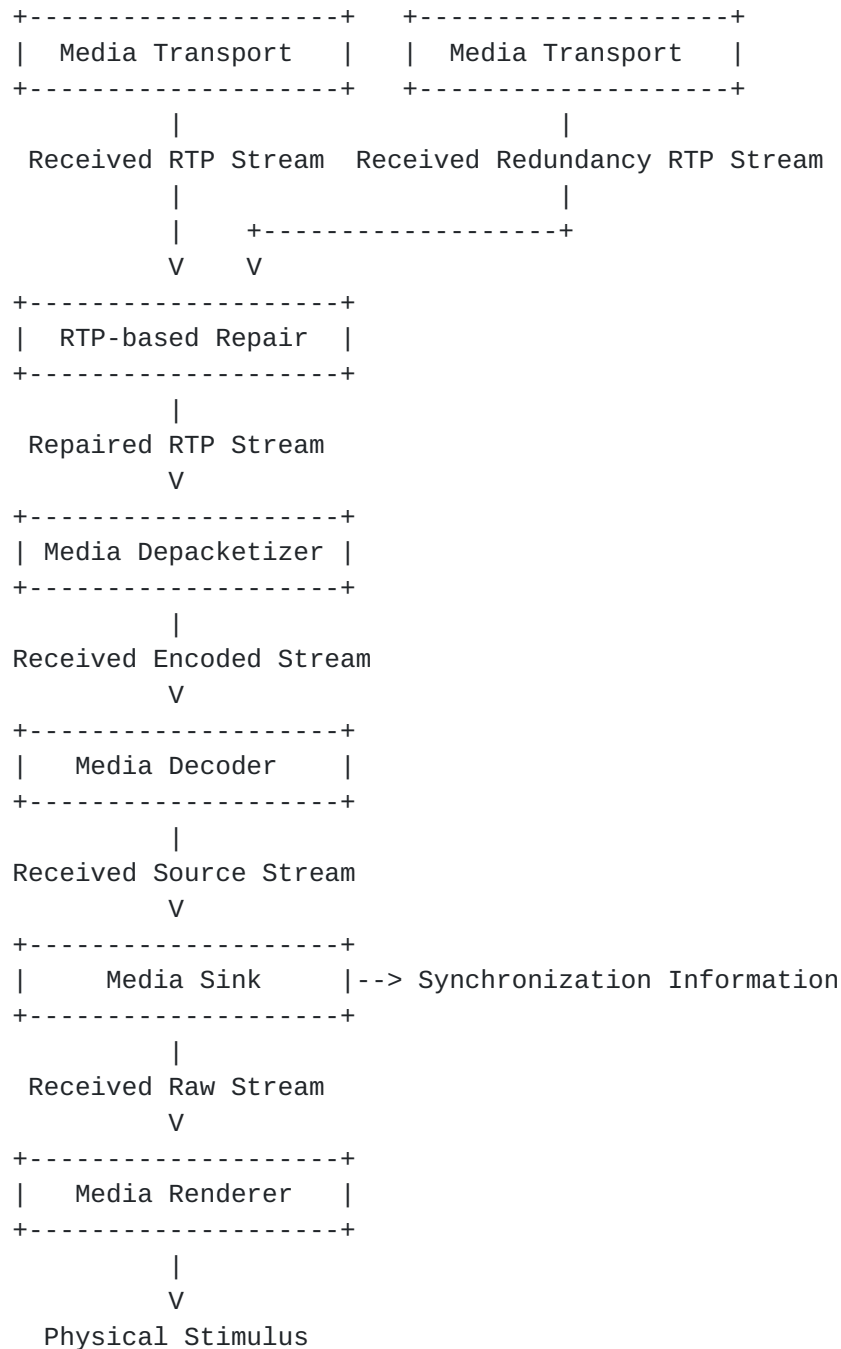This include sound waves making up audio, photons in a light field,
or other excitations or interactions with sensors, like keystrokes on
a keyboard.

2.1.2. **Media Capture**

Media Capture is the process of transforming the Physical Stimulus
(Section 2.1.1) into digital Media using an appropriate sensor or
transducer. The Media Capture performs a digital sampling of the
physical stimulus, usually periodically, and outputs this in some
representation as a Raw Stream (Section 2.1.3). This data is due to
its periodical sampling, or at least being timed asynchronous events,
some form of a stream of media data. The Media Capture is normally
instantiated in some type of device, i.e. media capture device.
Examples of different types of media capturing devices are digital
cameras, microphones connected to A/D converters, or keyboards.

Characteristics:

o A Media Capture is identified either by hardware/manufacturer ID
   or via a session-scoped device identifier as mandated by the
   application usage.

o A Media Capture can generate an Encoded Stream (Section 2.1.7) if
   the capture device support such a configuration.

o The nature of the Media Capture may impose constraints on the
   clock handling in some of the subsequent steps. For example, many
   audio or video capture devices are not completely free in
   selecting the sample rate.

2.1.3. **Raw Stream**

The time progressing stream of digitally sampled information, usually
periodically sampled and provided by a Media Capture (Section 2.1.2).
A Raw Stream can also contain synthesized Media that may not require
any explicit Media Capture, since it is already in an appropriate
digital form.

2.1.4. **Media Source**

A Media Source is the logical source of a reference clock
synchronized, time progressing, digital media stream, called a Source
Stream (Section 2.1.5). This transformation takes one or more Raw

Streams ([Section 2.1.3](#)) and provides a Source Stream as output. The output is synchronized with a reference clock ([Section 3.1](#)), which can be as simple as a system local wall clock or as complex as NTP synchronized.

The output can be of different types. One type is directly associated with a particular Media Capture's Raw Stream. Others are more conceptual sources, like an audio mix of multiple Source Streams (Figure 3). Mixing multiple streams typically requires that the input streams are possible to relate in time, meaning that they have to be Source Streams ([Section 2.1.5](#)) rather than Raw Streams. In the below example, the generated Source Stream is a mix of the three input Source Streams.

```
        Source      Source      Source
        Stream      Stream      Stream
          |           |           |
          V           V           V
     +---------------------------+
     |        Media Source       |<-- Reference Clock
     |            Mixer          |
     +---------------------------+
                   |
                   V
            Source Stream
```
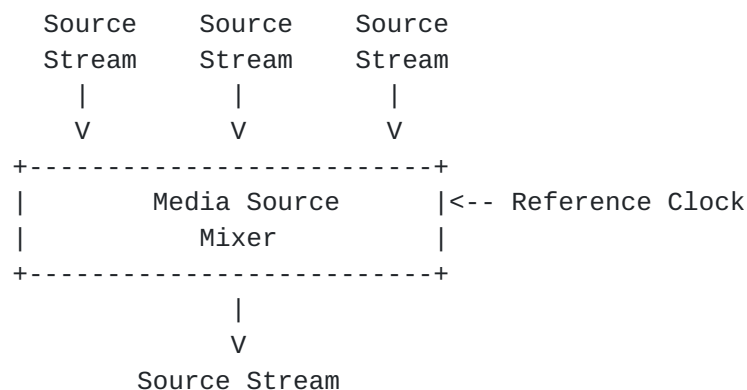
Figure 3: Conceptual Media Source in form of Audio Mixer

Another possible example of a conceptual Media Source is a video surveillance switch, where the input is multiple Source Streams from different cameras, and the output is one of those Source Streams based on some selection criteria, like a round-robin or based on some video activity measure.

Characteristics:

o At any point, it can represent a physical captured source or conceptual source.

2.1.5. Source Stream

A time progressing stream of digital samples that has been synchronized with a reference clock and comes from particular Media Source ([Section 2.1.4](#)).

2.1.6. Media Encoder

   A Media Encoder is a transform that is responsible for encoding the
   media data from a Source Stream (Section 2.1.5) into another
   representation, usually more compact, that is output as an Encoded
   Stream (Section 2.1.7).

   The Media Encoder step commonly includes pre-encoding
   transformations, such as scaling, resampling etc. The Media Encoder
   can have a significant number of configuration options that affects
   the properties of the Encoded Stream. This include properties such
   as bit-rate, start points for decoding, resolution, bandwidth or
   other fidelity affecting properties. The actually used codec is also
   an important factor in many communication systems.

   Scalable Media Encoders need special attention as they produce
   multiple outputs that are potentially of different types. A scalable
   Media Encoder takes one input Source Stream and encodes it into
   multiple output streams of two different types; at least one Encoded
   Stream that is independently decodable and one or more Dependent
   Streams (Section 2.1.8). Decoding requires at least one Encoded
   Stream and zero or more Dependent Streams. A Dependent Stream's
   dependency is one of the grouping relations this document discusses
   further in Section 3.8.

```
                        Source Stream
                             |
                             V
                +--------------------------+
                |  Scalable Media Encoder  |
                +--------------------------+
                    |       |   ...    |
                    V       V          V
                Encoded  Dependent  Dependent
                Stream    Stream     Stream
```
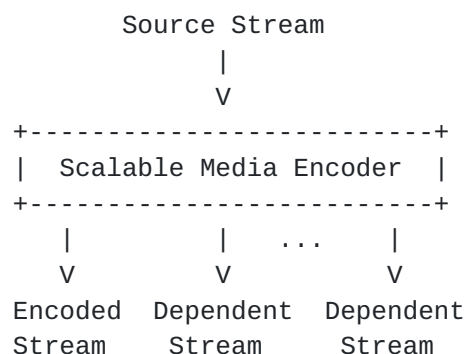
            Figure 4: Scalable Media Encoder Input and Outputs

   There are also other variants of encoders, like so-called Multiple
   Description Coding (MDC). Such Media Encoder produce multiple
   independent and thus individually decodable Encoded Streams.
   However, (logically) combining multiple of these Encoded Streams into
   a single Received Source Stream during decoding leads to an
   improvement in perceptual reproduced quality when compared to
   decoding a single Encoded Stream.

Creating multiple Encoded Streams from the same Source Stream, where
the Encoded Streams are neither in a scalable nor in an MDC
relationship is commonly utilized in Simulcast environments.

Characteristics:

o A Media Source can be multiply encoded by different Media Encoders
   to provide various encoded representations.

2.1.7. Encoded Stream

A stream of time synchronized encoded media that can be independently
decoded.

Characteristics:

o Due to temporal dependencies, an Encoded Stream may have
   limitations in where decoding can be started. These entry points,
   for example Intra frames from a video encoder, may require
   identification and their generation may be event based or
   configured to occur periodically.

2.1.8. Dependent Stream

A stream of time synchronized encoded media fragments that are
dependent on one or more Encoded Streams (Section 2.1.7) and zero or
more Dependent Streams to be possible to decode.

Characteristics:

o Each Dependent Stream has a set of dependencies. These
   dependencies must be understood by the parties in a Multimedia
   Session that intend to use a Dependent Stream.

2.1.9. Media Packetizer

The transformation of taking one or more Encoded (Section 2.1.7) or
Dependent Streams (Section 2.1.8) and put their content into one or
more sequences of packets, normally RTP packets, and output Source
RTP Streams (Section 2.1.10). This step includes both generating RTP
payloads as well as RTP packets.

The Media Packetizer can use multiple inputs when producing a single
RTP Stream. One such example is SRST packetization when using SVC
(Section 3.5).

The Media Packetizer can also produce multiple RTP Streams, for
example when Encoded and/or Dependent Streams are distributed over

multiple RTP Streams. One example of this is MRMT packetization when using SVC (Section 3.5).

Characteristics:

o The Media Packetizer will select which Synchronization source(s) (SSRC) [RFC3550] in which RTP Sessions that are used.

o Media Packetizer can combine multiple Encoded or Dependent Streams into one or more RTP Streams.

2.1.10. RTP Stream

A stream of RTP packets containing media data, source or redundant. The RTP Stream is identified by an SSRC belonging to a particular RTP Session. The RTP Session is identified as discussed in Section 2.2.2.

A Source RTP Stream is a RTP Stream containing at least some content from an Encoded Stream (Section 2.1.7). Source material is any media material that is produced for transport over RTP without any additional RTP-based redundancy applied. Note that RTP-based redundancy excludes the type of redundancy that most suitable Media Encoders (Section 2.1.6) may add to the media format of the Encoded Stream that makes it cope better with inevitable RTP packet losses. This is further described in RTP-based Redundancy (Section 2.1.11) and Redundancy RTP Stream (Section 2.1.12).

Characteristics:

o Each RTP Stream is identified by a Synchronization source (SSRC) [RFC3550] that is carried in every RTP and RTP Control Protocol (RTCP) packet header. The SSRC is unique in a specific RTP Session context.

o At any given point in time, a RTP Stream can have one and only one SSRC, but SSRCs for a given RTP Stream can change over time. SSRC collision and clock rate change [RFC7160] are examples of valid reasons to change SSRC for an RTP Stream. In those cases, the RTP Stream itself is not changed in any significant way, only the identifying SSRC number.

o Each SSRC defines a unique RTP sequence numbering and timing space.

o Several RTP Streams, each with their own SSRC, may represent a single Media Source.

   o Several RTP Streams, each with their own SSRC, can be carried in a
      single RTP Session.

2.1.11. **RTP-based Redundancy**

   RTP-based Redundancy is defined here as a transformation that
   generates redundant or repair packets sent out as a Redundancy RTP
   Stream (Section 2.1.12) to mitigate network transport impairments,
   like packet loss and delay.

   The RTP-based Redundancy exists in many flavors; they may be
   generating independent Repair Streams that are used in addition to
   the Source Stream (like RTP Retransmission (Section 3.11) and some
   special types of Forward Error Correction, like RTP stream
   duplication (Section 3.9)), they may generate a new Source Stream by
   combining redundancy information with source information (Using XOR
   FEC (Section 3.12) as a redundancy payload (Section 3.10)), or
   completely replace the source information with only redundancy
   packets.

2.1.12. **Redundancy RTP Stream**

   A RTP Stream (Section 2.1.10) that contains no original source data,
   only redundant data that may be combined with one or more Received
   RTP Stream (Section 2.1.19) to produce Repaired RTP Streams
   (Section 2.1.22).

2.1.13. **Media Transport**

   A Media Transport defines the transformation that the RTP Streams
   (Section 2.1.10) are subjected to by the end-to-end transport from
   one RTP sender to one specific RTP receiver (an RTP Session
   (Section 2.2.2) may contain multiple RTP receivers per sender). Each
   Media Transport is defined by a transport association that is
   normally identified by a 5-tuple (source address, source port,
   destination address, destination port, transport protocol), but a
   proposal exists for sending multiple transport associations on a
   single 5-tuple [I-D.westerlund-avtcore-transport-multiplexing].

   Characteristics:

   o Media Transport transmits RTP Streams of RTP Packets from a source
      transport address to a destination transport address.

   o Each Media Transport contains only a single RTP Session.

   o A single RTP Session can span multiple Media Transports.

The Media Transport concept sometimes needs to be decomposed into
more steps to enable discussion of what a sender emits that gets
transformed by the network before it is received by the receiver.
Thus we provide also this Media Transport decomposition (Figure 5).

```
                           RTP Stream
                               |
                               V
               +--------------------------+
               |  Media Transport Sender  |
               +--------------------------+
                              |
                       Sent RTP Stream
                              V
               +--------------------------+
               |     Network Transport    |
               +--------------------------+
                              |
                     Transported RTP Stream
                              V
               +--------------------------+
               | Media Transport Receiver |
               +--------------------------+
                              |
                              V
                     Received RTP Stream
```
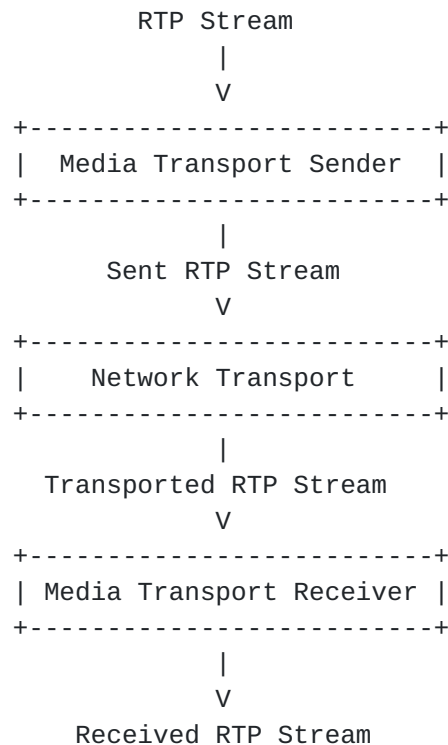
Figure 5: Decomposition of Media Transport

2.1.14. Media Transport Sender

The first transformation within the Media Transport (Section 2.1.13)
is the Media Transport Sender. The sending Endpoint (Section 2.2.1)
takes an RTP Stream and emits the packets onto the network using the
transport association established for this Media Transport, thereby
creating a Sent RTP Stream (Section 2.1.15). In the process, it
transforms the RTP Stream in several ways. First, it generates the
necessary protocol headers for the transport association, for example
IP and UDP headers, thus forming IP/UDP/RTP packets. In addition,
the Media Transport Sender may queue, pace or otherwise affect how
the packets are emitted onto the network, thereby potentially
introducing delay, jitter and inter packet spacings that characterize
the Sent RTP Stream.

2.1.15. Sent RTP Stream

   The Sent RTP Stream is the RTP Stream as entering the first hop of
   the network path to its destination. The Sent RTP Stream is
   identified using network transport addresses, like for IP/UDP the
   5-tuple (source IP address, source port, destination IP address,
   destination port, and protocol (UDP)).

2.1.16. Network Transport

   Network Transport is the transformation that subjects the Sent RTP
   Stream (Section 2.1.15) to traveling from the source to the
   destination through the network. This transformation can result in
   loss of some packets, varying delay on a per packet basis, packet
   duplication, and packet header or data corruption. This
   transformation produces a Transported RTP Stream (Section 2.1.17) at
   the exit of the network path.

2.1.17. Transported RTP Stream

   The RTP Stream that is emitted out of the network path at the
   destination, subjected to the Network Transport's transformation
   (Section 2.1.16).

2.1.18. Media Transport Receiver

   The receiver Endpoint's (Section 2.2.1) transformation of the
   Transported RTP Stream (Section 2.1.17) by its reception process,
   which results in the Received RTP Stream (Section 2.1.19). This
   transformation includes transport checksums being verified. Sensible
   system designs typically either discard packets with mis-matching
   checksums, or pass them on while somehow marking them in the
   resulting Received RTP Stream so to alarm subsequent transformations
   about the possible corrupt state. In this context it is worth noting
   that there is typically some probability for corrupt packets to pass
   through undetected (with a seemingly correct checksum). Other
   transformations can compensate for delay variations in receiving a
   packet on the network interface and providing it to the application
   (de-jitter buffer).

2.1.19. Received RTP Stream

   The RTP Stream (Section 2.1.10) resulting from the Media Transport's
   transformation, i.e. subjected to packet loss, packet corruption,
   packet duplication and varying transmission delay from sender to
   receiver.

2.1.20. **Received Redundancy RTP Stream**

   The Redundancy RTP Stream (Section 2.1.12) resulting from the Media
   Transport transformation, i.e. subjected to packet loss, packet
   corruption, and varying transmission delay from sender to receiver.

2.1.21. **RTP-based Repair**

   RTP-based Repair is a Transformation that takes as input one or more
   Received RTP Streams (Section 2.1.19) and Received Redundancy RTP
   Streams (Section 2.1.20), and produces one or more Repaired RTP
   Streams (Section 2.1.22) that are as close to the corresponding sent
   Source RTP Streams (Section 2.1.10) as possible, using different RTP-
   based repair methods, for example the ones referred in RTP-based
   Redundancy (Section 2.1.11).

2.1.22. **Repaired RTP Stream**

   A Received RTP Stream (Section 2.1.19) for which Received Redundancy
   RTP Stream (Section 2.1.20) information has been used to try to
   recover the Source RTP Stream (Section 2.1.10) as it was before Media
   Transport (Section 2.1.13).

2.1.23. **Media Depacketizer**

   A Media Depacketizer takes one or more RTP Streams (Section 2.1.10),
   depacketizes them, and attempts to reconstitute the Encoded Streams
   (Section 2.1.7) or Dependent Streams (Section 2.1.8) present in those
   RTP Streams.

   It should be noted that in practical implementations, the Media
   Depacketizer and the Media Decoder may be tightly coupled and share
   information to improve or optimize the overall decoding and error
   concealment process. It is, however, not expected that there would
   be any benefit in defining a taxonomy for those detailed (and likely
   very implementation-dependent) steps.

2.1.24. **Received Encoded Stream**

   The received version of an Encoded Stream (Section 2.1.7).

2.1.25. **Media Decoder**

   A Media Decoder is a transformation that is responsible for decoding
   Encoded Streams (Section 2.1.7) and any Dependent Streams
   (Section 2.1.8) into a Source Stream (Section 2.1.5).

It should be noted that in practical implementations, the Media
Decoder and the Media Depacketizer may be tightly coupled and share
information to improve or optimize the overall decoding process in
various ways. It is however not expected that there would be any
benefit in defining a taxonomy for those detailed (and likely very
implementation-dependent) steps.

Characteristics:

o A Media Decoder has to deal with any errors in the Encoded Streams
   that resulted from corruption or failure to repair packet losses.
   Therefore, it commonly is robust to error and losses, and includes
   concealment methods.

2.1.26. Received Source Stream

The received version of a Source Stream (Section 2.1.5).

2.1.27. Media Sink

The Media Sink receives a Source Stream (Section 2.1.5) that
contains, usually periodically, sampled media data together with
associated synchronization information. Depending on application,
this Source Stream then needs to be transformed into a Raw Stream
(Section 2.1.3) that is conveyed to the Media Render
(Section 2.1.29), synchronized with the output from other Media
Sinks. The Media Sink may also be connected with a Media Source
(Section 2.1.4) and be used as part of a conceptual Media Source.

Characteristics:

o The Media Sink can further transform the Source Stream into a
   representation that is suitable for rendering on the Media Render
   as defined by the application or system-wide configuration. This
   include sample scaling, level adjustments etc.

2.1.28. Received Raw Stream

The received version of a Raw Stream (Section 2.1.3).

2.1.29. Media Render

A Media Render takes a Raw Stream (Section 2.1.3) and converts it
into Physical Stimulus (Section 2.1.1) that a human user can
perceive. Examples of such devices are screens, and D/A converters
connected to amplifiers and loudspeakers.

Characteristics:

   o An Endpoint can potentially have multiple Media Renders for each
      media type.

2.2. Communication Entities

   This section contains concept for entities involved in the
   communication.

```
   +-----------------------------------------------------------------+
   | Communication Session                                           |
   |                                                                 |
   | +---------------+                        +---------------+ |
   | | Participant A |     +------------+     | Participant B | |
   | |               |     | Multimedia |     |               | |
   | | +-------------+|<==>| Session    |<==>|+-------------+ | |
   | | | Endpoint A  ||    |            |     || Endpoint B  | | |
   | | |             ||    +------------+     ||             | | |
   | | | +----------++--------------------++----------+ | | |
   | | | |          |                      |          | | | |
   | | | | RTP Session|---Media Transport--->|         | | | |
   | | | | Audio      |<---Media Transport---|         | | | |
   | | | |          |          ^           |          | | | |
   | | | +----------++---------|----------++----------+ | | |
   | | |          ||          v           ||          | | |
   | | |          || +----------------+   ||          | | |
   | | |          || | Synchronization |  ||          | | |
   | | |          || |    Context      |  ||          | | |
   | | |          || +----------------+   ||          | | |
   | | |          ||          ^           ||          | | |
   | | | +----------++---------|----------++----------+ | | |
   | | | |          |          v           |          | | | |
   | | | | RTP Session|<---Media Transport---|        | | | |
   | | | | Video      |---Media Transport--->|        | | | |
   | | | |          |                      |          | | | |
   | | | +----------++--------------------++----------+ | | |
   | | +-------------+|                      |+-------------+ | |
   | +---------------+                        +---------------+ |
   +-----------------------------------------------------------------+
```
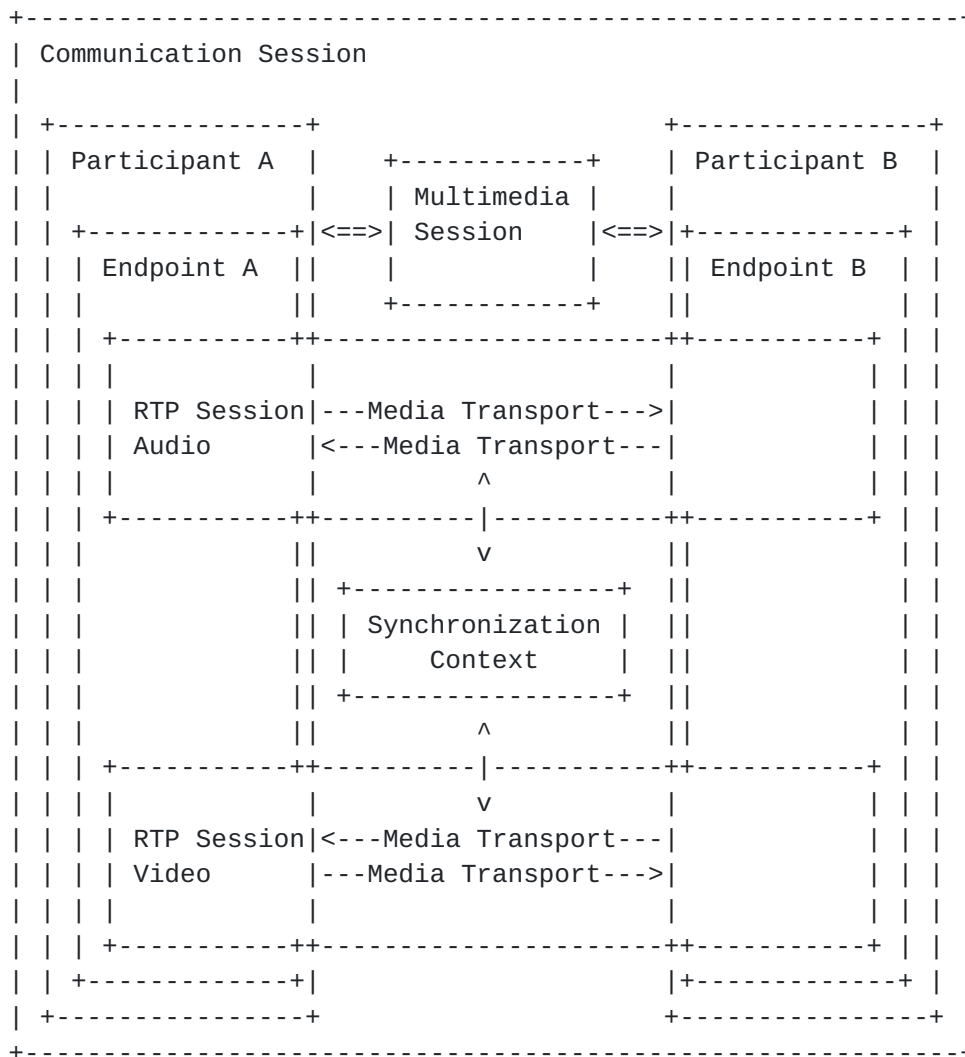
          Figure 6: Example Point to Point Communication Session with two RTP
                                Sessions

   The figure above shows a high-level example representation of a very
   basic point-to-point Communication Session between Participants A and
   B. It uses two different audio and video RTP Sessions between A's
   and B's Endpoints, using separate Media Transports for those RTP
   Sessions. The Multimedia Session shared by the Participants can, for
   example, be established using SIP (i.e., there is a SIP Dialog

between A and B). The terms used in that figure are further
elaborated in the sub-sections below.

2.2.1. Endpoint

A single addressable entity sending or receiving RTP packets. It may
be decomposed into several functional blocks, but as long as it
behaves as a single RTP stack entity it is classified as a single
"Endpoint".

Characteristics:

o Endpoints can be identified in several different ways. While RTCP
   Canonical Names (CNAMEs) [RFC3550] provide a globally unique and
   stable identification mechanism for the duration of the
   Communication Session (see Section 2.2.5), their validity applies
   exclusively within a Synchronization Context (Section 3.1). Thus
   one Endpoint can handle multiple CNAMEs, each of which can be
   shared among a set of Endpoints belonging to the same Participant
   (Section 2.2.3). Therefore, mechanisms outside the scope of RTP,
   such as application defined mechanisms, must be used to ensure
   Endpoint identification when outside this Synchronization Context.

o An Endpoint can be associated with at most one Participant
   (Section 2.2.3) at any single point in time.

o In some contexts, an Endpoint would typically correspond to a
   single "host", for example a computer using a single network
   interface and being used by a single human user.

2.2.2. RTP Session

An RTP Session is an association among a group of Participants
communicating with RTP. It is a group communications channel which
can potentially carry a number of RTP Streams. Within an RTP
Session, every Participant can find meta-data and control information
(over RTCP) about all the RTP Streams in the RTP Session. The
bandwidth of the RTCP control channel is shared between all
Participants within an RTP Session.

Characteristics:

o An RTP Session can carry one ore more RTP Streams.

o An RTP Session shares a single SSRC space as defined in RFC3550
   [RFC3550]. That is, the Endpoints participating in an RTP Session
   can see an SSRC identifier transmitted by any of the other
   Endpoints. An Endpoint can receive an SSRC either as SSRC or as a

Contributing source (CSRC) in RTP and RTCP packets, as defined by
the Endpoints' network interconnection topology.

o An RTP Session uses at least two Media Transports
   (Section 2.1.13), one for sending and one for receiving.
   Commonly, the receiving Media Transport is the reverse direction
   of the Media Transport used for sending. An RTP Session may use
   many Media Transports and these define the session's network
   interconnection topology.

o A single Media Transport always carries a single RTP Session.

o Multiple RTP Sessions can be conceptually related, for example
   originating from or targeted for the same Participant
   (Section 2.2.3) or Endpoint (Section 2.2.1), or by containing RTP
   Streams that are somehow related (Section 3).

2.2.3. Participant

A Participant is an entity reachable by a single signaling address,
and is thus related more to the signaling context than to the media
context.

Characteristics:

o A single signaling-addressable entity, using an application-
   specific signaling address space, for example a SIP URI.

o A Participant can participate in several Multimedia Sessions
   (Section 2.2.4).

o A Participant can be comprised of several associated Endpoints
   (Section 2.2.1).

2.2.4. Multimedia Session

A Multimedia Session is an association among a group of Participants
(Section 2.2.3) engaged in the communication via one or more RTP
Sessions (Section 2.2.2). It defines logical relationships among
Media Sources (Section 2.1.4) that appear in multiple RTP Sessions.

Characteristics:

o A Multimedia Session can be composed of several RTP Sessions with
   potentially multiple RTP Streams per RTP Session.

o Each Participant in a Multimedia Session can have a multitude of
   Media Captures and Media Rendering devices.

o A single Multimedia Session can contain media from one or more
   Synchronization Contexts ([Section 3.1](#)). An example of that is a
   Multimedia Session containing one set of audio and video for
   communication purposes belonging to one Synchronization Context,
   and another set of audio and video for presentation purposes (like
   playing a video file) with a separate Synchronization Context that
   has no strong timing relationship and need not be strictly
   synchronized with the audio and video used for communication.

2.2.5. Communication Session

A Communication Session is an association among two or more
Participants ([Section 2.2.3](#)) communicating with each other via one or
more Multimedia Sessions ([Section 2.2.4](#)).

Characteristics:

o Each Participant in a Communication Session is identified via an
   application-specific signaling address.

o A Communication Session is composed of Participants that share at
   least one Multimedia Session, involving one or more parallel RTP
   Sessions with potentially multiple RTP Streams per RTP Session.

For example, in a full mesh communication, the Communication Session
consists of a set of separate Multimedia Sessions between each pair
of Participants. Another example is a centralized conference, where
the Communication Session consists of a set of Multimedia Sessions
between each Participant and the conference handler.

3. Concepts of Inter-Relations

This section uses the concepts from previous sections, and looks at
different types of relationships among them. These relationships
occur at different abstraction levels and for different purposes, but
the reason for the needed relationship at a certain step in the media
handling chain may exist at another step. For example, the use of
Simulcast ([Section 3.7](#))) implies a need to determine relations at RTP
Stream level, but the underlying reason is that multiple Media
Encoders use the same Media Source, i.e. to be able to identify a
common Media Source.

3.1. Synchronization Context

A Synchronization Context defines a requirement on a strong timing
relationship between the Media Sources, typically requiring alignment
of clock sources. Such a relationship can be identified in multiple
ways as listed below. A single Media Source can only belong to a

single Synchronization Context, since it is assumed that a single
Media Source can only have a single media clock and requiring
alignment to several Synchronization Contexts (and thus reference
clocks) will effectively merge those into a single Synchronization
Context.

3.1.1. RTCP CNAME

RFC3550 [RFC3550] describes Inter-media synchronization between RTP
Sessions based on RTCP CNAME, RTP and Network Time Protocol (NTP)
[RFC5905] formatted timestamps of a reference clock. As indicated in
[RFC7273], despite using NTP format timestamps, it is not required
that the clock be synchronized to an NTP source.

3.1.2. Clock Source Signaling

[RFC7273] provides a mechanism to signal the clock source in SDP both
for the reference clock as well as the media clock, thus allowing a
Synchronization Context to be defined beyond the one defined by the
usage of CNAME source descriptions.

3.1.3. Implicitly via RtcMediaStream

The WebRTC WG defines "RtcMediaStream" with one or more
"RtcMediaStreamTracks". All tracks in a "RtcMediaStream" are
intended to be synchronized when rendered, implying that they must be
generated such that synchronization is possible.

3.1.4. Explicitly via SDP Mechanisms

The SDP Grouping Framework [RFC5888] defines an m= line (Section 4.2)
grouping mechanism called "Lip Synchronization (LS)" for establishing
the synchronization requirement across m= lines when they map to
individual sources.

Source-Specific Media Attributes in SDP [RFC5576] extends the above
mechanism when multiple Media Sources are described by a single m=
line.

3.2. Endpoint

Some applications requires knowledge of what Media Sources originate
from a particular Endpoint (Section 2.2.1). This can include such
decisions as packet routing between parts of the topology, knowing
the Endpoint origin of the RTP Streams.

In RTP, this identification has been overloaded with the
Synchronization Context (Section 3.1) through the usage of the RTCP

source description CNAME ([Section 3.1.1](#)). This works for some
usages, but in others it breaks down. For example, if an Endpoint
has two sets of Media Sources that have different Synchronization
Contexts, like the audio and video of the human Participant as well
as a set of Media Sources of audio and video for a shared movie,
CNAME would not be an appropriate identification for that Endpoint.
Therefore, an Endpoint may have multiple CNAMEs. The CNAMEs or the
Media Sources themselves can be related to the Endpoint.

3.3. Participant

In communication scenarios, it is commonly needed to know which Media
Sources originate from which Participant ([Section 2.2.3](#)). One reason
is, for example, to enable the application to display Participant
Identity information correctly associated with the Media Sources.
This association is handled through the signaling solution to point
at a specific Multimedia Session where the Media Sources may be
explicitly or implicitly tied to a particular Endpoint.

Participant information becomes more problematic due to Media Sources
that are generated through mixing or other conceptual processing of
Raw Streams or Source Streams that originate from different
Participants. This type of Media Sources can thus have a dynamically
varying set of origins and Participants. RTP contains the concept of
Contributing Sources (CSRC) that carry information about the previous
step origin of the included media content on RTP level.

3.4. RtcMediaStream

An RtcMediaStream in WebRTC is an explicit grouping of a set of Media
Sources (RtcMediaStreamTracks) that share a common identifier and a
single Synchronization Context ([Section 3.1](#)).

3.5. Single- and Multi-Session Transmission of Dependent Streams

Scalable media coding formats such as, for example, H.264 based
Scalable Video Coding [[RFC6190](#)] has two modes of operation:

1. In Single Session Transmission (SST), the SVC Media Encoder sends
    Encoded Streams ([Section 2.1.7](#)) and Dependent Streams
    ([Section 2.1.8](#)) as a single RTP Stream ([Section 2.1.10](#)) in a
    single RTP Session ([Section 2.2.2](#)), using the SVC RTP Payload
    format.

2. In Multi-Session Transmission (MST), the SVC Media Encoder sends
    Encoded Streams and Dependent Streams distributed across multiple
    RTP Streams in one or more RTP Sessions.

SST denotes one RTP Stream (SSRC) per Media Source in a single RTP
Session. MST denotes one or more RTP Streams (SSRC) per Media Source
in each of multiple RTP Sessions. The above is not unambiguously
specified in the SVC payload format text [RFC6190], but it is what
existing deployments of that RFC have implemented.

The use of the term "RTP Session" in the SST/MST definition is
somewhat misleading, since a single RTP Session can contain multiple
RTP Streams. Also, it is sometimes useful to make a distinction
between using a single Media Transport or multiple separate Media
Transports when (in both cases) using multiple RTP Streams to carry
Encoded Streams and Dependent Streams for a Media Source. Therefore,
herein the following new terminology is defined:

SRST: Single RTP Stream on a Single Media Transport

MRST: Multiple RTP Streams on a Single Media Transport

MRMT: Multiple RTP Streams on Multiple Media Transports

3.6. Multi-Channel Audio

There exist a number of RTP payload formats that can carry multi-
channel audio, despite the codec being a mono encoder. Multi-channel
audio can be viewed as multiple Media Sources sharing a common
Synchronization Context. These are independently encoded by a Media
Encoder and the different Encoded Streams are packetized together in
a time synchronized way into a single Source RTP Stream, using the
used codec's RTP Payload format. Examples of codecs that support
multi-channel audio are PCMA and PCMU [RFC3551], AMR [RFC4867], and
G.719 [RFC5404].

3.7. Simulcast

A Media Source represented as multiple independent Encoded Streams
constitutes a Simulcast or Multiple Description Coding of that Media
Source. Figure 7 below shows an example of a Media Source that is
encoded into three separate Simulcast streams, that are in turn sent
on the same Media Transport flow. When using Simulcast, the RTP
Streams may be sharing RTP Session and Media Transport, or be
separated on different RTP Sessions and Media Transports, or any
combination of these two. It is other considerations that affect
which usage is desirable, as discussed in Section 3.13.

```
                         +----------------+
                         |  Media Source  |
                         +----------------+
                  Source Stream  |
           +--------------------+--------------------+
           |                    |                    |
           V                    V                    V
   +-----------------+  +-----------------+  +-----------------+
   |  Media Encoder  |  |  Media Encoder  |  |  Media Encoder  |
   +-----------------+  +-----------------+  +-----------------+
            | Encoded            | Encoded            | Encoded
            | Stream             | Stream             | Stream
            V                    V                    V
   +-----------------+  +-----------------+  +-----------------+
   | Media Packetizer|  | Media Packetizer|  | Media Packetizer|
   +-----------------+  +-----------------+  +-----------------+
            | Source             | Source             | Source
            | RTP                | RTP                | RTP
            | Stream             | Stream             | Stream
            +-----------------+  |     +-----------------+
                              |  |  |
                              V  V  V
                      +------------------+
                      |  Media Transport |
                      +------------------+
```
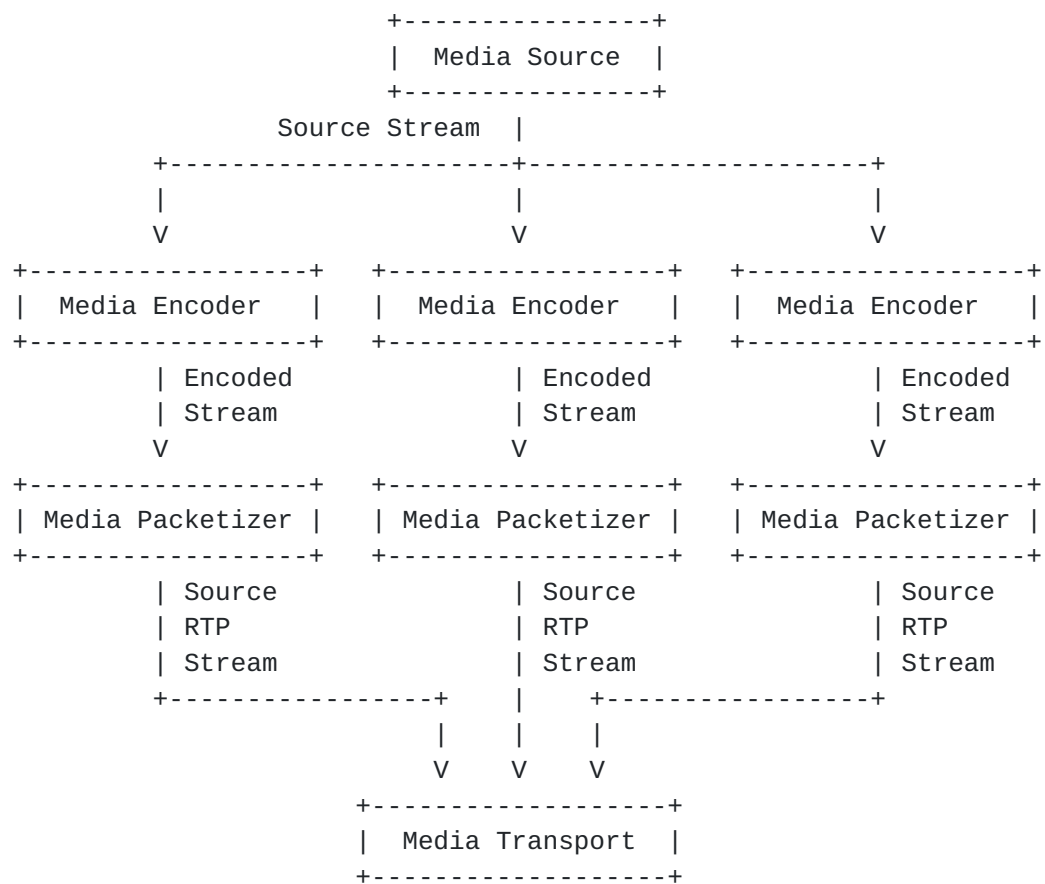
            Figure 7: Example of Media Source Simulcast

   The Simulcast relation between the RTP Streams is the common Media
   Source. In addition, to be able to identify the common Media Source,
   a receiver of the RTP Stream may need to know which configuration or
   encoding goals that lay behind the produced Encoded Stream and its
   properties. This to enable selection of the stream that is most
   useful in the application at that moment.

3.8. Layered Multi-Stream

   Layered Multi-Stream (LMS) is a mechanism by which different portions
   of a layered encoding of a Source Stream are sent using separate RTP
   Streams (sometimes in separate RTP Sessions). LMSs are useful for
   receiver control of layered media.

   A Media Source represented as an Encoded Stream and multiple
   Dependent Streams constitutes a Media Source that has layered
   dependencies. The figure below represents an example of a Media
   Source that is encoded into three dependent layers, where two layers
   are sent on the same Media Transport using different RTP Streams,

i.e. SSRCs, and the third layer is sent on a separate Media
Transport.

```
                        +----------------+
                        |  Media Source  |
                        +----------------+
                                |
                                |
                                V
     +------------------------------------------------------------+
     |                      Media Encoder                         |
     +------------------------------------------------------------+
            |                  |                    |
       Encoded Stream    Dependent Stream     Dependent Stream
            |                  |                    |
            V                  V                    V
     +----------------+ +----------------+  +----------------+
     |Media Packetizer| |Media Packetizer|  |Media Packetizer|
     +----------------+ +----------------+  +----------------+
            |                  |                    |
        RTP Stream         RTP Stream           RTP Stream
            |                  |                    |
        +------+          +------+                 |
            |          |                           |
            V          V                           V
         +-----------------+          +-----------------+
         | Media Transport |          | Media Transport |
         +-----------------+          +-----------------+
```
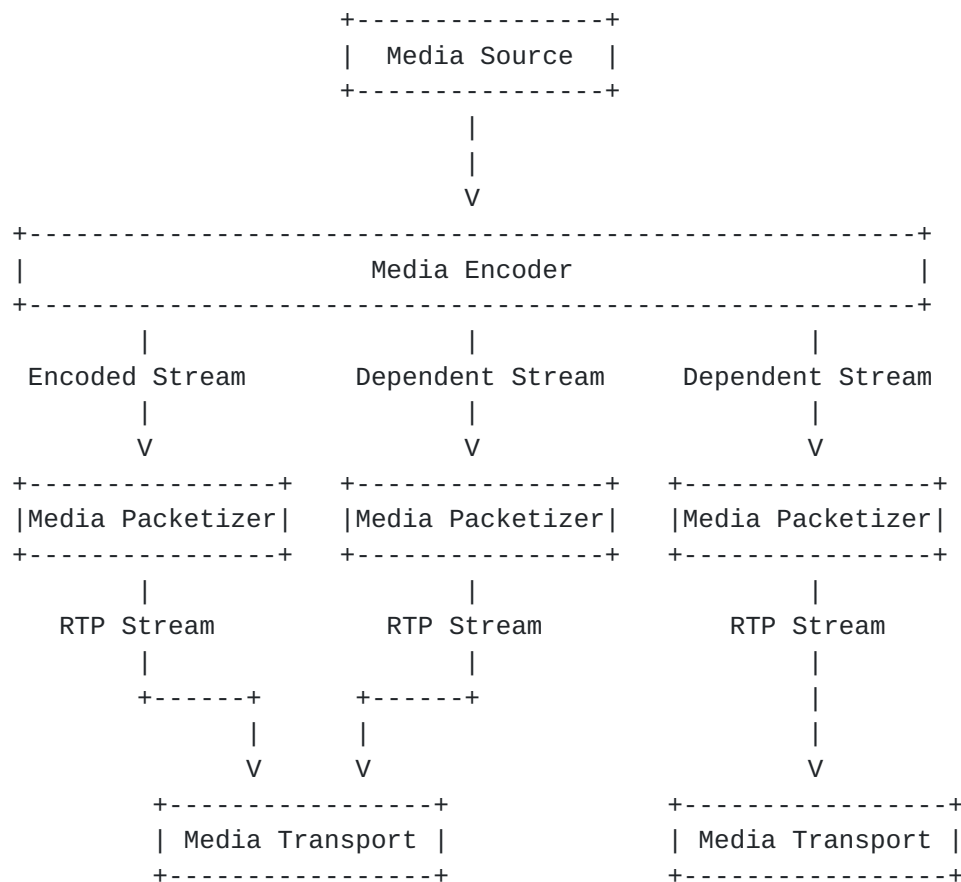
Figure 8: Example of Media Source Layered Dependency

As an example, the SVC MRST and MRMT (Section 3.5) relations needs to
identify the common Media Encoder origin for the Encoded and
Dependent Streams. The SVC RTP Payload RFC [RFC6190] is not
particularly explicit about how this relation is to be implemented.
When using different RTP Sessions, thus different Media Transports
(MRMT (Section 3.5)), and as long as there is only one RTP Stream per
Media Encoder and a single Media Source in each RTP Session (MRMT),
common SSRC and CNAMEs can be used to identify the common Media
Source. When multiple RTP Streams are sent from one Media Encoder in
the same RTP Session (MRST), then CNAME is the only currently
specified RTP identifier that can be used. In cases where multiple
Media Encoders use multiple Media Sources sharing Synchronization
Context, and thus having a common CNAME, additional heuristics or
identification need to be applied to create the MRST or MRMT
relationships between the RTP Streams.

3.9. RTP Stream Duplication

RTP Stream Duplication [RFC7198], using the same or different Media
Transports, and optionally also delaying the duplicate [RFC7197],
offers a simple way to protect media flows from packet loss in some
cases. It is a specific type of redundancy and all but one Source
RTP Stream (Section 2.1.10) are effectively Redundancy RTP Streams
(Section 2.1.12), but since both Source and Redundant RTP Streams are
the same it does not matter which one is which. This can also be
seen as a specific type of Simulcast (Section 3.7) that transmits the
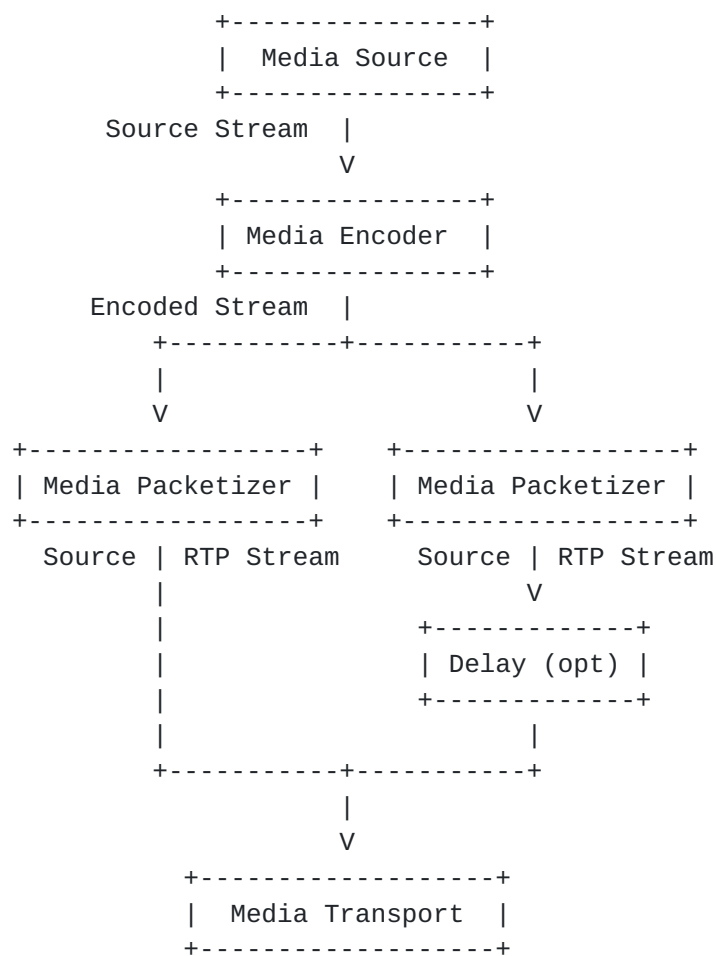same Encoded Stream (Section 2.1.7) multiple times.

```
                        +----------------+
                        |  Media Source  |
                        +----------------+
                Source Stream  |
                               V
                        +----------------+
                        | Media Encoder  |
                        +----------------+
                Encoded Stream  |
                    +-----------+-----------+
                    |                       |
                    V                       V
            +------------------+    +------------------+
            | Media Packetizer |    | Media Packetizer |
            +------------------+    +------------------+
              Source | RTP Stream     Source | RTP Stream
                     |                       V
                     |               +-------------+
                     |               | Delay (opt) |
                     |               +-------------+
                     |                       |
                    +-----------+-----------+
                                |
                                V
                        +------------------+
                        |  Media Transport |
                        +------------------+
```

Figure 9: Example of RTP Stream Duplication

3.10. Redundancy Format

The RTP Payload for Redundant Audio Data [RFC2198] defines a
transport for redundant audio data together with primary data in the
same RTP payload. The redundant data can be a time delayed version
of the primary or another time delayed Encoded Stream using a

different Media Encoder to encode the same Media Source as the
primary, as depicted below in Figure 10.

```
             +--------------------+
             |    Media Source    |
             +--------------------+
                       |
                 Source Stream
                       |
                 +------------------------+
                 |                        |
                 V                        V
       +--------------------+   +--------------------+
       |   Media Encoder    |   |   Media Encoder    |
       +--------------------+   +--------------------+
                 |                        |
                 |              +------------+
            Encoded Stream      | Time Delay |
                 |              +------------+
                 |                        |
                 |      +------------------+
                 V      V
       +--------------------+
       |  Media Packetizer  |
       +--------------------+
                 |
                 V
             RTP Stream
```
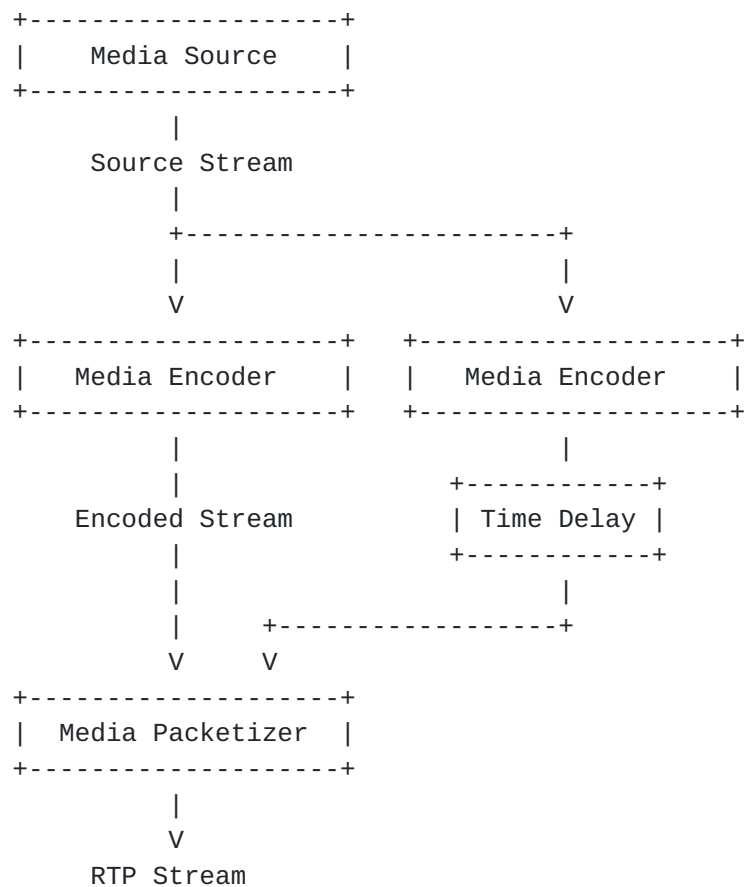
Figure 10: Concept for usage of Audio Redundancy with different Media
Encoders

The Redundancy format is thus providing the necessary meta
information to correctly relate different parts of the same Encoded
Stream, or in the case depicted above (Figure 10) relate the Received
Source Stream fragments coming out of different Media Decoders to be
able to combine them together into a less erroneous Source Stream.

3.11. RTP Retransmission

Figure 11 shows an example where a Media Source's Source RTP Stream
is protected by a retransmission (RTX) flow [RFC4588]. In this
example the Source RTP Stream and the Redundancy RTP Stream share the
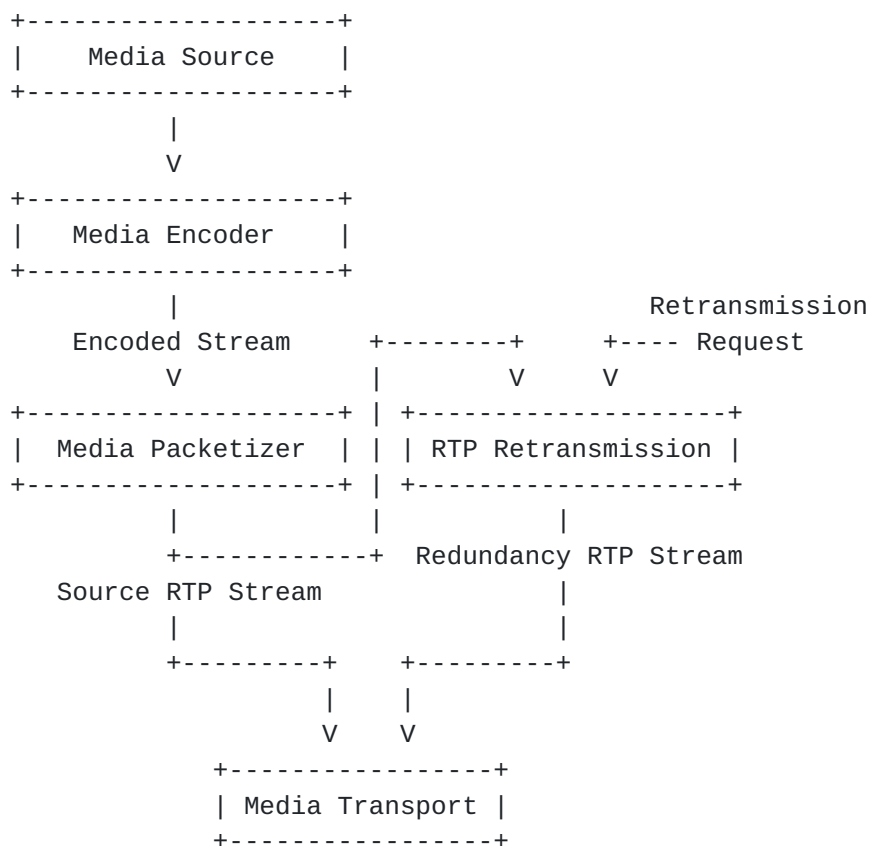same Media Transport.

```
           +--------------------+
           |    Media Source    |
           +--------------------+
                     |
                     V
           +--------------------+
           |   Media Encoder    |
           +--------------------+
                     |                            Retransmission
             Encoded Stream     +--------+     +---- Request
                     V          |        V     V
           +--------------------+ | +--------------------+
           |  Media Packetizer  | | | RTP Retransmission |
           +--------------------+ | +--------------------+
                     |            |          |
                     +-----------+  Redundancy RTP Stream
               Source RTP Stream              |
                     |                        |
                     +---------+    +---------+
                          |    |
                          V    V
                   +-----------------+
                   | Media Transport |
                   +-----------------+
```

Figure 11: Example of Media Source Retransmission Flows

The RTP Retransmission example (Figure 11) illustrates that this
mechanism works purely on the Source RTP Stream. The RTP
Retransmission transform buffers the sent Source RTP Stream and, upon
request, emits a retransmitted packet with an extra payload header as
a Redundancy RTP Stream. The RTP Retransmission mechanism [RFC4588]
is specified such that there is a one to one relation between the
Source RTP Stream and the Redundancy RTP Stream. Therefore, a
Redundancy RTP Stream needs to be associated with its Source RTP
Stream. This is done based on CNAME selectors and heuristics to
match requested packets for a given Source RTP Stream with the
original sequence number in the payload of any new Redundancy RTP
Stream using the RTX payload format. In cases where the Redundancy
RTP Stream is sent in a separate RTP Session from the Source RTP
Stream, these sessions are related, which is signaled by using the
SDP Media Grouping's [RFC5888] FID semantics.

3.12. Forward Error Correction

The figure below (Figure 12) shows an example where two Media
Sources' Source RTP Streams are protected by FEC. Source RTP Stream
A has a RTP-based Redundancy transformation in FEC Encoder 1. This

produces a Redundancy RTP Stream 1, that is only related to Source
RTP Stream A. The FEC Encoder 2, however, takes two Source RTP
Streams (A and B) and produces a Redundancy RTP Stream 2 that
protects them jointly, i.e. Redundancy RTP Stream 2 relates to two
Source RTP Streams (a FEC group). FEC decoding, when needed due to
packet loss or packet corruption at the receiver, requires knowledge
about which Source RTP Streams that the FEC encoding was based on.

In Figure 12 all RTP Streams are sent on the same Media Transport.
This is however not the only possible choice. Numerous combinations
exist for spreading these RTP Streams over different Media Transports
to achieve the communication application's goal.

```
   +--------------------+                +--------------------+
   |   Media Source A   |                |   Media Source B   |
   +--------------------+                +--------------------+
             |                                     |
             V                                     V
   +--------------------+                +--------------------+
   |   Media Encoder A  |                |   Media Encoder B  |
   +--------------------+                +--------------------+
             |                                     |
        Encoded Stream                        Encoded Stream
             V                                     V
   +--------------------+                +--------------------+
   | Media Packetizer A |                | Media Packetizer B |
   +--------------------+                +--------------------+
             |                                     |
      Source RTP Stream A                   Source RTP Stream B
             |                                     |
      +-----+---------+------------+        +---+---+
      |               V            V        V   |
      |         +--------------+  +--------------+    |
      |         | FEC Encoder 1 |  | FEC Encoder 2 |    |
      |         +--------------+  +--------------+    |
      |     Redundancy   |      Redundancy   |        |
      |     RTP Stream 1 |      RTP Stream 2 |        |
      V               V                  V        V
   +------------------------------------------------------------+
   |                     Media Transport                        |
   +------------------------------------------------------------+
```
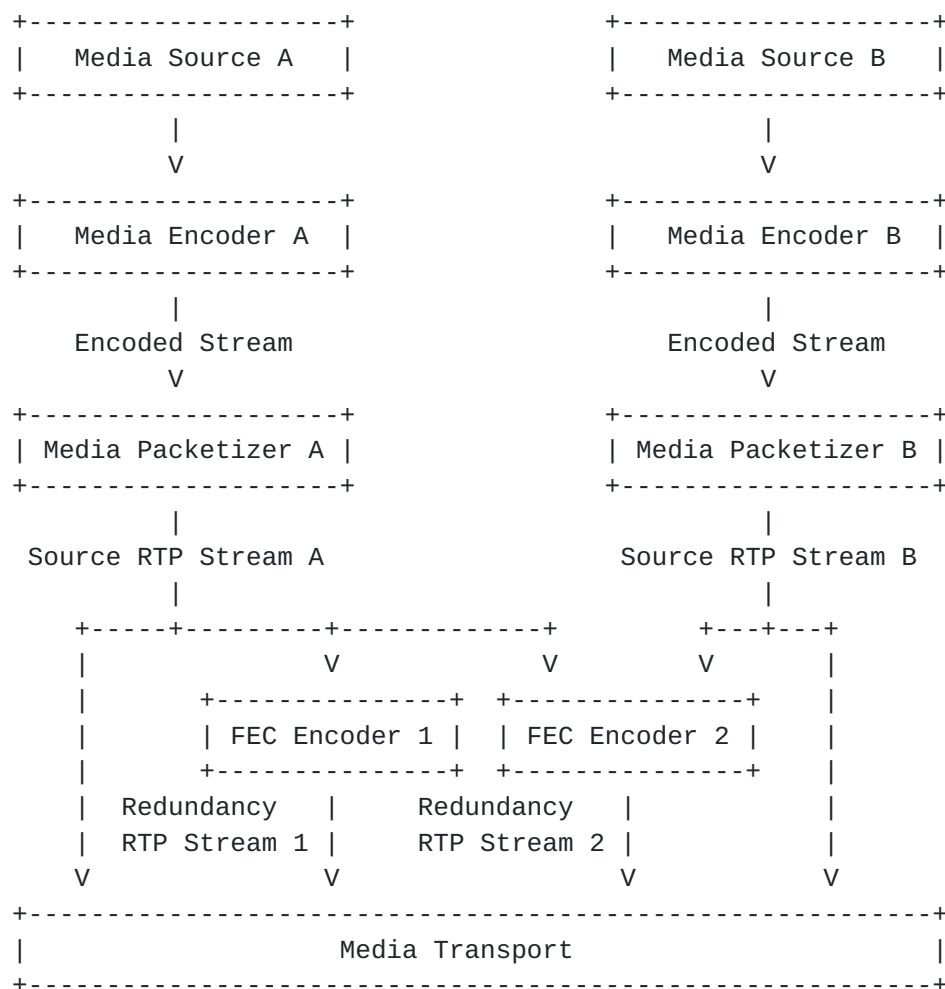
              Figure 12: Example of FEC Redundancy RTP Streams

As FEC Encoding exists in various forms, the methods for relating FEC
Redundancy RTP Streams with its source information in Source RTP
Streams are many. The XOR based RTP FEC Payload format [RFC5109] is
defined in such a way that a Redundancy RTP Stream has a one to one

relation with a Source RTP Stream. In fact, the RFC requires the
Redundancy RTP Stream to use the same SSRC as the Source RTP Stream.
This requires to either use a separate RTP Session or to use the
Redundancy RTP Payload format [RFC2198]. The underlying relation
requirement for this FEC format and a particular Redundancy RTP
Stream is to know the related Source RTP Stream, including its SSRC.

3.13. RTP Stream Separation

RTP Streams can be separated exclusively based on their SSRCs, at the
RTP Session level, or at the Multi-Media Session level.

When the RTP Streams that have a relationship are all sent in the
same RTP Session and are uniquely identified based on their SSRC
only, it is termed an SSRC-Only Based Separation. Such streams can
be related via RTCP CNAME to identify that the streams belong to the
same Endpoint. SSRC-based approaches [RFC5576], when used, can
explicitly relate various such RTP Streams.

On the other hand, when RTP Streams that are related but are sent in
the context of different RTP Sessions to achieve separation, it is
known as RTP Session-based separation. This is commonly used when
the different RTP Streams are intended for different Media
Transports.

Several mechanisms that use RTP Session-based separation rely on it
to enable an implicit grouping mechanism expressing the relationship.
The solutions have been based on using the same SSRC value in the
different RTP Sessions to implicitly indicate their relation. That
way, no explicit RTP level mechanism has been needed, only signaling
level relations have been established using semantics from Grouping
of Media lines framework [RFC5888]. Examples of this are RTP
Retransmission [RFC4588], SVC Multi-Session Transmission [RFC6190]
and XOR Based FEC [RFC5109]. RTCP CNAME explicitly relates RTP
Streams across different RTP Sessions, as explained in the previous
section. Such a relationship can be used to perform inter-media
synchronization.

RTP Streams that are related and need to be associated can be part of
different Multimedia Sessions, rather than just different RTP
Sessions within the same Multimedia Session context. This puts
further demand on the scope of the mechanism(s) and its handling of
identifiers used for expressing the relationships.

3.14. Multiple RTP Sessions over one Media Transport

   [I-D.westerlund-avtcore-transport-multiplexing] describes a mechanism
   that allows several RTP Sessions to be carried over a single
   underlying Media Transport. The main reasons for doing this are
   related to the impact of using one or more Media Transports (using a
   common network path or potentially have different ones). The fewer
   Media Transports used, the less need for NAT/FW traversal resources
   and number of flow based QoS.

   However, Multiple RTP Sessions over one Media Transport imply that a
   single Media Transport 5-tuple is not sufficient to express in which
   RTP Session context a particular RTP Stream exists. Complexities in
   the relationship between Media Transports and RTP Session already
   exist as one RTP Session contains multiple Media Transports, e.g.
   even a Peer-to-Peer RTP Session with RTP/RTCP Multiplexing requires
   two Media Transports, one in each direction. The relationship
   between Media Transports and RTP Sessions as well as additional
   levels of identifiers need to be considered in both signaling design
   and when defining terminology.

4. Mapping from Existing Terms

   This section describes a selected set of terms from some relevant
   IETF RFC and Internet Drafts (at the time of writing), using the
   concepts from previous sections.

4.1. Telepresence Terms

   The terms in this sub-section are used in the context of CLUE
   Telepresence [I-D.ietf-clue-framework].

4.1.1. Audio Capture

   Describes an audio Media Source (Section 2.1.4).

4.1.2. Capture Device

   Identifies a physical entity performing a Media Capture
   (Section 2.1.2) transformation.

4.1.3. Capture Encoding

   Describes an Encoded Stream (Section 2.1.7) related to CLUE specific
   semantic information.

4.1.4. Capture Scene

Describes a set of spatially related Media Sources (Section 2.1.4).

4.1.5. Endpoint

Describes exactly one Participant (Section 2.2.3) and one or more
Endpoints (Section 2.2.1).

4.1.6. Individual Encoding

Describes the configuration information needed to perform a Media
Encoder (Section 2.1.6) transformation.

4.1.7. Media Capture

Describes either a Media Capture (Section 2.1.2) or a Media Source
(Section 2.1.4), depending on in which context the term is used.

4.1.8. Media Consumer

Describes the media receiving part of an Endpoint (Section 2.2.1).

4.1.9. Media Provider

Describes the media sending part of an Endpoint (Section 2.2.1).

4.1.10. Stream

Describes an RTP Stream (Section 2.1.10).

4.1.11. Video Capture

Describes a video Media Source (Section 2.1.4).

4.2. Media Description

A single Source Description Protocol (SDP) [RFC4566] media
description (or media block; an m-line and all subsequent lines until
the next m-line or the end of the SDP) describes part of the
necessary configuration and identification information needed for a
Media Encoder transformation, as well as the necessary configuration
and identification information for the Media Decoder to be able to
correctly interpret a received RTP Stream.

A Media Description typically relates to a single Media Source. This
is for example an explicit restriction in WebRTC. However, nothing
prevents that the same Media Description (and same RTP Session) is

re-used for multiple Media Sources
[I-D.ietf-avtcore-rtp-multi-stream]. It can thus describe properties
of one or more RTP Streams, and can also describe properties valid
for an entire RTP Session (via [RFC5576] mechanisms, for example).

4.3. Media Stream

RTP [RFC3550] uses media stream, audio stream, video stream, and
stream of (RTP) packets interchangeably, which are all RTP Streams.

4.4. Multimedia Conference

A Multimedia Conference is a Communication Session (Section 2.2.5)
between two or more Participants (Section 2.2.3), along with the
software they are using to communicate.

4.5. Multimedia Session

SDP [RFC4566] defines a Multimedia Session as a set of multimedia
senders and receivers and the data streams flowing from senders to
receivers, which would correspond to a set of Endpoints and the RTP
Streams that flow between them. In this memo, Multimedia Session
(Section 2.2.4) also assumes those Endpoints belong to a set of
Participants that are engaged in communication via a set of related
RTP Streams.

RTP [RFC3550] defines a Multimedia Session as a set of concurrent RTP
Sessions among a common group of Participants. For example, a video
conference may contain an audio RTP Session and a video RTP Session.
This would correspond to a group of Participants (each using one or
more Endpoints) sharing a set of concurrent RTP Sessions. In this
memo, Multimedia Session also defines those RTP Sessions to have some
relation and be part of a communication among the Participants.

4.6. Multipoint Control Unit (MCU)

This term is commonly used to describe the central node in any type
of star topology [I-D.ietf-avtcore-rtp-topologies-update] conference.
It describes a device that includes one Participant (Section 2.2.3)
(usually corresponding to a so-called conference focus) and one or
more related Endpoints (Section 2.2.1) (sometimes one or more per
conference Participant).

4.7. Recording Device

WebRTC specifications use this term to refer to locally available
entities performing a Media Capture (Section 2.1.2) transformation.

4.8. RtcMediaStream

A WebRTC RtcMediaStreamTrack is a set of Media Sources
(Section 2.1.4) sharing the same Synchronization Context
(Section 3.1).

4.9. RtcMediaStreamTrack

A WebRTC RtcMediaStreamTrack is a Media Source (Section 2.1.4).

4.10. RTP Sender

RTP [RFC3550] uses this term, which can be seen as the RTP protocol
part of a Media Packetizer (Section 2.1.9).

4.11. RTP Session

Within the context of SDP, a singe m= line can map to a single RTP
Session (Section 2.2.2) or multiple m= lines can map to a single RTP
Session. The latter is enabled via multiplexing schemes such as
BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], for example, which
allows mapping of multiple m= lines to a single RTP Session.

4.12. SSRC

RTP [RFC3550] defines this as "the source of a stream of RTP
packets", which indicates that an SSRC is not only a unique
identifier for the Encoded Stream (Section 2.1.7) carried in those
packets, but is also effectively used as a term to denote a Media
Packetizer (Section 2.1.9).

5. Security Considerations

This document simply tries to clarify the confusion prevalent in RTP
taxonomy because of inconsistent usage by multiple technologies and
protocols making use of the RTP protocol. It does not introduce any
new security considerations beyond those already well documented in
the RTP protocol [RFC3550] and each of the many respective
specifications of the various protocols making use of it.

Hopefully having a well-defined common terminology and understanding
of the complexities of the RTP architecture will help lead us to
better standards, avoiding security problems.

6. Acknowledgement

   This document has many concepts borrowed from several documents such
   as WebRTC [I-D.ietf-rtcweb-overview], CLUE [I-D.ietf-clue-framework],
   Multiplexing Architecture
   [I-D.westerlund-avtcore-transport-multiplexing]. The authors would
   like to thank all the authors of each of those documents.

   The authors would also like to acknowledge the insights, guidance and
   contributions of Magnus Westerlund, Roni Even, Paul Kyzivat, Colin
   Perkins, Keith Drage, Harald Alvestrand, Alex Eleftheriadis, Mo
   Zanaty, and Stephan Wenger.

7. Contributors

   Magnus Westerlund has contributed the concept model for the media
   chain using transformations and streams model, including rewriting
   pre-existing concepts into this model and adding missing concepts.
   The first proposal for updating the relationships and the topologies
   based on this concept was also performed by Magnus.

8. IANA Considerations

   This document makes no request of IANA.

9. Informative References

   [I-D.ietf-avtcore-rtp-multi-stream]
             Lennox, J., Westerlund, M., Wu, W., and C. Perkins,
             "Sending Multiple Media Streams in a Single RTP Session",
             draft-ietf-avtcore-rtp-multi-stream-06 (work in progress),
             October 2014.

   [I-D.ietf-avtcore-rtp-topologies-update]
             Westerlund, M. and S. Wenger, "RTP Topologies", draft-
             ietf-avtcore-rtp-topologies-update-05 (work in progress),
             November 2014.

   [I-D.ietf-clue-framework]
             Duckworth, M., Pepperell, A., and S. Wenger, "Framework
             for Telepresence Multi-Streams", draft-ietf-clue-
             framework-19 (work in progress), December 2014.

   [I-D.ietf-mmusic-sdp-bundle-negotiation]
             Holmberg, C., Alvestrand, H., and C. Jennings,
             "Negotiating Media Multiplexing Using the Session
             Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-
             negotiation-14 (work in progress), December 2014.

[I-D.ietf-rtcweb-overview]
          Alvestrand, H., "Overview: Real Time Protocols for
          Browser-based Applications", draft-ietf-rtcweb-overview-13
          (work in progress), November 2014.

[I-D.westerlund-avtcore-transport-multiplexing]
          Westerlund, M. and C. Perkins, "Multiplexing Multiple RTP
          Sessions onto a Single Lower-Layer Transport", draft-
          westerlund-avtcore-transport-multiplexing-07 (work in
          progress), October 2013.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V.,
          Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-
          Parisis, "RTP Payload for Redundant Audio Data", RFC 2198,
          September 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
          Jacobson, "RTP: A Transport Protocol for Real-Time
          Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and
          Video Conferences with Minimal Control", STD 65, RFC 3551,
          July 2003.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
          Description Protocol", RFC 4566, July 2006.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R.
          Hakenberg, "RTP Retransmission Payload Format", RFC 4588,
          July 2006.

[RFC4867] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie,
          "RTP Payload Format and File Storage Format for the
          Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband
          (AMR-WB) Audio Codecs", RFC 4867, April 2007.

[RFC5109] Li, A., "RTP Payload Format for Generic Forward Error
          Correction", RFC 5109, December 2007.

[RFC5404] Westerlund, M. and I. Johansson, "RTP Payload Format for
          G.719", RFC 5404, January 2009.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific
          Media Attributes in the Session Description Protocol
          (SDP)", RFC 5576, June 2009.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description
          Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

   [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network
              Time Protocol Version 4: Protocol and Algorithms
              Specification", RFC 5905, June 2010.

   [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis,
              "RTP Payload Format for Scalable Video Coding", RFC 6190,
              May 2011.

   [RFC7160] Petit-Huguenin, M. and G. Zorn, "Support for Multiple
              Clock Rates in an RTP Session", RFC 7160, April 2014.

   [RFC7197] Begen, A., Cai, Y., and H. Ou, "Duplication Delay
              Attribute in the Session Description Protocol", RFC 7197,
              April 2014.

   [RFC7198] Begen, A. and C. Perkins, "Duplicating RTP Streams", RFC
              7198, April 2014.

   [RFC7273] Williams, A., Gross, K., van Brandenburg, R., and H.
              Stokking, "RTP Clock Source Signalling", RFC 7273, June
              2014.

Appendix A. Changes From Earlier Versions

   NOTE TO RFC EDITOR: Please remove this section prior to publication.

A.1. Modifications Between WG Version -03 and -04

   o Changed "Media Redundancy" and "Media Repair" to "RTP-based
      Redundancy" and "RTP-based Repair", since those terms are more
      specific and correct.

   o Changed "End Point" to "Endpoint" and removed Editor's Note on
      this.

   o Clarified that a Media Capture may impose constraints on clock
      handling.

   o Clarified that mixing multiple Raw Streams into a Source Stream is
      not possible, since that requires mixed streams to have a timing
      relation, requiring them to be Source Streams, and added an
      example.

   o Clarified that RTP-based Redundancy excludes the type of encoding
      redundancy found within the encoded media format in an Encoded
      Stream.

o Clarified that a Media Transport contains only a single RTP
   Session, but a single RTP Session can span multiple Media
   Transports.

o Clarified that packets with seemingly correct checksum that are
   received by a Media Transport Receiver may still be corrupt.

o Clarified that a corrupt packet in a Media Transport Receiver is
   typically either discarded or somehow marked and passed on in the
   Received RTP Stream.

o Added Synchronization Context to Figure 6.

o Editorial improvements and clarifications.

A.2. Modifications Between WG Version -02 and -03

o Changed section 3.5, removing SST-SS/MS and MST-SS/MS, replacing
   them with SRST, MRST, and MRMT.

o Updated section 3.8 to align with terminology changes in section
   3.5.

o Added a new section 4.12, describing the term Multimedia
   Conference.

o Changed reference from I-D to now published RFC 7273.

o Editorial improvements and clarifications.

A.3. Modifications Between WG Version -01 and -02

o Major re-structure

o Moved media chain Media Transport detailing up one section level

o Collapsed level 2 sub-sections of section 3 and thus moved level 3
   sub-sections up one level, gathering some introductory text into
   the beginning of section 3

o Added that not only SSRC collision, but also a clock rate change
   [RFC7160] is a valid reason to change SSRC value for an RTP stream

o Added a sub-section on clock source signaling

o Added a sub-section on RTP stream duplication

o Elaborated a bit in section 2.2.1 on the relation between End
   Points, Participants and CNAMEs

o Elaborated a bit in section 2.2.4 on Multimedia Session and
   synchronization contexts

o Removed the section on CLUE scenes defining an implicit
   synchronization context, since it was incorrect

o Clarified text on SVC SST and MST according to list discussions

o Removed the entire topology section to avoid possible
   inconsistencies or duplications with draft-ietf-avtcore-rtp-
   topologies-update, but saved one example overview figure of
   Communication Entities into that section

o Added a section 4 on mapping from existing terms with one sub-
   section per term, mainly by moving text from sections 2 and 3

o Changed all occurrences of Packet Stream to RTP Stream

o Moved all normative references to informative, since this is an
   informative document

o Added references to RFC 7160, RFC 7197 and RFC 7198, and removed
   unused references

A.4. Modifications Between WG Version -00 and -01

o WG version -00 text is identical to individual draft -03

o Amended description of SVC SST and MST encodings with respect to
   concepts defined in this text

o Removed UML as normative reference, since the text no longer uses
   any UML notation

o Removed a number of level 4 sections and moved out text to the
   level above

A.5. Modifications Between Version -02 and -03

o Section 4 rewritten (and new communication topologies added) to
   reflect the major updates to Sections 1-3

o Section 8 removed (carryover from initial -00 draft)

o General clean up of text, grammar and nits

A.6. **Modifications Between Version -01 and -02**

o Section 2 rewritten to add both streams and transformations in the
   media chain.

o Section 3 rewritten to focus on exposing relationships.

A.7. **Modifications Between Version -00 and -01**

o Too many to list

o Added new authors

o Updated content organization and presentation

Authors' Addresses

   Jonathan Lennox
   Vidyo, Inc.
   433 Hackensack Avenue
   Seventh Floor
   Hackensack, NJ 07601
   US

   Email: jonathan@vidyo.com

   Kevin Gross
   AVA Networks, LLC
   Boulder, CO
   US

   Email: kevin.gross@avanw.com

   Suhas Nandakumar
   Cisco Systems
   170 West Tasman Drive
   San Jose, CA 95134
   US

   Email: snandaku@cisco.com

Gonzalo Salgueiro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: gsalguei@cisco.com

Bo Burman
Ericsson
Kistavagen 25
SE-164 80 Stockholm
Sweden

Phone: +46 10 714 13 11
Email: bo.burman@ericsson.com